CENTRE
DE RECHERCHE
SUR LES TRANSPORTS
(CRT)
CENTRE
FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

C.P. 6128, SUCCURSALE A
MONTRÉAL (QUÉBEC)
CANADA
H3C 3J7

TÉLÉPHONE : (514) 343-7575
TÉLEX : 05-24146 BIBPOLYTEC
TÉLÉCOPIEUR (FAX) : (514) 343-7121

CRT-647

VEHICLE CREW SCHEDULING TO COMPLETE SPECIFIC TASKS AND BULK-TASKS AT DEPOTS

by

Jacques A. Ferland
and
Serge Taillefer

This research was supported by the N.S.E.R.C. (Grant A8312).

Université de Montréal

Centre de recherche sur les transports - Publication # 647
Laboratoire d'optimisation du DIRO - Publication # LO 89-5
Département d'Informatique
et de Recherche opérationnelle - Publication # 681

April 1989

90-01411/

Dépôt légal - 3e trimestre 1989

Abstract: In this vehicle crew scheduling problem, we consider several depots, and each operator is based at one of these depots. The workday of each operator is specified in terms of specific tasks and a share of the bulk-task of his depot. The model is formulated as a set-covering problem with additional constraints. The solution approach includes three major steps. First, we generate several reasonable schedules with regard to union contract regulations. Then the corresponding set-covering problem with additional constraints is solved with a branch-and-bound procedure. Finally, this solution is modified and improved by reorganizing the sharing of the bulk-tasks among operators. Numerical results are reported for problems in context of postal vehicle crew scheduling.

Keywords: Crew scheduling, set covering, branch-and-bound.

Résumé: Nous considérons un problème d'horaires de chauffeurs de véhicules où il y a plusieurs dépôts et où chaque chauffeur est rattaché à un dépôt. Les responsabilités journalières de chaque chauffeur sont spécifiées en terme de tâches spécifiques et d'une répartition d'une tâche globale qui incombe à son dépôt. Le modèle est formulé comme un problème de recouvrement avec des contraintes additionnelles. L'approche de résolution comporte trois étapes majeures. La première consiste à générer un ensemble d'horaires respectant certaines contraintes de la convention collective des chauffeurs. Par la suite, nous générons un problème de recouvrement avec des contraintes additionnelles pour assurer la couverture de la tâche globale des dépôts et la proportion de chauffeurs réguliers requise par la convention collective. Nous utilisons une procédure de type "branch-and-bound" pour traiter ce problème. Finalement, la solution obtenue est améliorée en répartissant plus adéquatement la tâche globale du dépôt entre ses chauffeurs. Des résultats numériques sont présentés dans le contexte du problème des horaires des chauffeurs de véhicules de transport du courrier.

Mots clés: Horaires de chauffeurs, problème de recouvrement, "branch-and-bound".

1. INTRODUCTION

In the vehicle crew scheduling problem analyzed in this paper, we consider several depots k ($1 \leq k \leq K$), and each operator is based at one of these depots. The workday of each operator is specified in terms of tasks executed.

Two types of tasks are considered in this problem. First, there is a set T of fixed tasks i , $1 \leq i \leq n$. For each depot k, we denote by $T_k \subset T$ the subset of tasks workable by its operators. Note that these subsets are not a partition of T since task i can belong to several of these.

Each task i is characterized as follows:

- starting location SL_i
- ending location EL_i
- duration D_i
- starting time ST_i
- time window $[a_i, b_i]$ for the starting time ST_i ; *i.e.* $a_i \leq ST_i \leq b_i$.

These parameters are known in advance.

Secondly, there are bulk-tasks associated with depots. Each of these is specified in terms of a specific volume of work (number of hours) to be shared arbitrarily among the operators of the depot.

In general, several types of operators are allowed, but in our implementation we restrict ourselves to two types: full-time operators and part-time operators. For each type, there is a guaranteed minimum number of daily working hours that are paid at regular rate, and any additional time, up to a specified maximum, is paid at overtime rate.

When the operator picks up and returns his vehicle, time is provided for preparation and report. Hence, the time available to execute tasks has to be adjusted accordingly.

With regard to union contract regulations, appropriate coffee-breaks and lunch-break have to be provided for the operators. Upper and lower bounds are specified for their length. Also, they can be partly or entirely paid. A full-time operator has a right to two coffee-breaks and a lunch-break, and a part-time operator, to one coffee-break. Furthermore, these regulations specify earliest and latest starting time for these breaks relatively to beginning of the operator's working period. Finally, in general, the union contract regulations limit the proportion of part-time operators.

The objective of this vehicle crew scheduling model is to determine the number of each type of operators required at each depot to execute all tasks at minimum cost. The model also accounts for union contract regulations related to breaks and proportion of part-time operators.

In Section 2, the vehicle crew scheduling model is formulated as a set-covering problem with additional constraints, and the solution approach is summarized. The procedure to generate reasonable daily schedules (with regard to union contract regulations) is presented in Section 3. In Section 4, we introduce the branch-and-bound method to solve the set-covering problem with additional constraints. A procedure to modify and improve the solution is introduced in Section 5. Finally, Section 6 includes concluding remarks on the implementation and the numerical results.

2. PROBLEM FORMULATION AND SOLUTION APPROACH

The vehicle crew scheduling model is formulated as a set-covering problem with additional constraints. The variables are associated with reasonable daily schedules with regard to union contract regulations. The additional constraints specify that an appropriate proportion of full-time operators is satisfied and that the bulk-task is completed at each depot.

The following notation is used in the formulation of the problem:

Ω^F : set of reasonable daily schedules for full-time operators

Ω^P : set of reasonable daily schedules for part-time operators

$\Omega = \Omega^F \cup \Omega^P$

c_j : cost of schedule $j \in \Omega$

$\delta_{ij} =$ $\delta_{ij} \begin{cases} 1 & \textit{if task } i \textit{ is included in schedule } j \\ 0 & \textit{otherwise,} \end{cases}$

$1 \le i \le n, j \in \Omega$

f_{kj} : length of time in schedule j available for the bulk-task at depot k ,

$1 \le k \le K, j \in \Omega$

w_k : volume of work (number of hours) required by bulk-task at depot k,

$1 \le k \le K$.

The decision variables are the following:

$$y_j = \begin{cases} 1 & \textit{if schedule } j \textit{ is selected} \\ 0 & \textit{otherwise,} \end{cases} \qquad j \in \Omega .$$

Then the vehicle crew scheduling model is formulated as follows:

(P) Min $\sum_{j \in \Omega} c_j y_j$

Subject to

$$\sum_{j \in \Omega} \delta_{ij} y_j \geq 1 \qquad 1 \leq i \leq n \qquad (2.1)$$

$$\sum_{j \in \Omega} f_{kj} y_j \geq w_k \qquad 1 \leq k \leq K \qquad (2.2)$$

$$(1 - \alpha) \sum_{j \in \Omega^F} y_j - \alpha \sum_{j \in \Omega^P} y_j \geq 0 \qquad (2.3)$$

$$y_j = 0 \textit{ or } 1 \qquad j \in \Omega$$

where α , $0 \leq \alpha \leq 1$, is the lower bound on the proportion of full-time operators; i.e.,

$$\frac{\sum_{j \in \Omega^F} y_j}{\sum_{j \in \Omega^F} y_j + \sum_{j \in \Omega^P} y_j} \geq \alpha .$$

Our approach to deal with the problem includes three major steps:

Step 1. **Daily schedules generation.** A set Ω of reasonable daily schedules with regard to union contract regulations is generated.

Step 2. **Solving problem (P).** Problem (P) is solved with a Branch-and-Bound method.

Step 3. **Modifying and improving the solution.** Starting with the feasible solution obtained in Step 3, we reorganize the sharing of the bulk-task at depots to reduce the total cost.

In the next sections, we present the details of the procedures used at each step. However, note that Step 3 is useful to induce additional saving by slightly reorganizing the schedules selected in Step 2 to reduce overtime.

3. REASONABLE DAILY SCHEDULES

Our approach is to generate a large variety of daily schedules for the different types of operators because it is unrealistic to generate all reasonable schedules. Indeed, since no regulations specify starting times of daily schedules, their number is rather large. Furthermore, even if no such regulation exists, it is expected in fact that most daily schedules should begin approximatively at two or three different specific times (8h30, 13h00, 17h00 for instance). In our implementation, such an assumption is made.

For each depot, the procedure is used to generate reasonable schedules including fixed tasks workable by its operators. The procedure is applied for each approximate starting time allowed for schedules, for each type of operator allowing for minimal and maximal time length schedule, respectively. Hence, we generate twelve different types of schedules for each depot.

The schedules generator procedure is a variant of the assignment method [3, 5] used to deal with single depot vehicle scheduling problems with time windows. This variant is accounting for the union contract regulations with regard to coffee and lunch breaks.

Assume that we are generating reasonable schedules starting approximatively at time τ_l ($1 \leq l \leq 3$) for depot k. To introduce the procedure, we refer to the following oriented graph (N,A):

i) With each task $i \in T_k$ such that $a_i \geq \tau_l$, associate a node $i \in N$. With depot k, associate two nodes s and t.

ii) An arc (i, j) exists (i.e., (i, j) ∈ A or task j can follow task i) if

$$a_i + D_i + t_{ij} \leq b_j$$

where t_{ij} is the time required to travel from EL_i to SL_j. Furthermore, for each task $i \in N$, arcs (s, i) and (i, t) exist.

In terms of this oriented graph, the procedure partitions the nodes into a set of paths (or reasonable daily schedules) starting at s and ending at t, so that the time window constraints are satisfied for each task $i \in N$ and the union contract regulations with regard to breaks are met. Furthermore, during the procedure, an appropriate cost structure is introduced to reduce the number of daily schedules required to cover all fixed tasks $i \in N$ and the length of time during which the operators are idle between two consecutive tasks of their schedules.

At each iteration of the procedure, a transportation problem is specified in terms of two subsets of tasks, N_1 and N_2. The subset N_1 is the set of source-tasks including those completed but not followed by any other. The subset N_2 is the set of sink-tasks to be executed next.

For the first iteration, the subset N_1 includes tasks in $N' = N - \{s, t\}$ that cannot be executed after any other task; i.e., $j \in N'$ if $a_i + D_i + t_{ij} > b_j$ for all $i \in N$, $i \neq j$. If no such task exists, then there are cycles of tasks that can follow each other but no other task. In this case, N_1 includes only one task i in one of these cycles with the smallest execution time D_i. The purpose of this strategy is to have the operator executing this task free as early as possible to execute other tasks. To determine N_2, replace N' by $N' - N_1$ and repeat the procedure used to specify N_1.

For each pair of nodes $i \in N_1$, $j \in N_2$, an arc (i, j) exists if

- task j can be executed after i without exceeding the length of time allowed for a schedule;
- appropriate breaks can be included in the resulting schedule.

If arc (i, j) exists, then its capacity is equal to 1 and the assignment cost to have task j following last i is given by

$$ac_{ij} = Max\ \{t_{ij},\ \bar{h}_j - h_i\}$$

where h_i is the completion time of task i in the schedule and $\bar{h}_j$ is the earliest time at which task j can begin considering its time window and appropriate break (if required). If fact ac_{ij} represents the time lost between tasks i and j. For each $i \in N_1$, there exists an arc (i, t) having a capacity of 1 and a cost $ac_{it} = 0$ to indicate that another task can be executed after i , later on. For each $j \in N_2$, there exists an arc (s, j) having a capacity of 1 and a cost $ac_{sj} = \mu$ (μ being a scalar taking a very large value) to indicate that an additional schedule is initiated with task j. Finally, arc (s, t) is introduced. Its capacity is not limited and $ac_{st} = 0$.

To complete the definition of the transportation problem associated with the iteration, the availabilities and demands are specified as follows:

s :	availability of $\|N_2\|$ units
$i \in N_1$:	availability of 1 unit
$j \in N_2$:	demand of 1 unit
t :	demand of $\|N_1\|$ units.

This transportation problem is solved. Denote by x, an optimal solution.

For $i \in N_1$, if $x_{ij} = 1$, then i remains in set N_1 to indicate that another task can be included in the schedule after i. For $j \in N_2$, if $x_{sj} = 1$, a new schedule is initiated with task j , and j is introduced in N_1. If $x_{ij} = 1$ $(i \neq s, j \neq t)$ then task j is included in a schedule after i. Then i is eliminated from N_1 and j is introduced in N_1. Furthermore the completion time h_j of task j is taken as early as possible with regard to its time window and appropriate break to be provided. Also, if a lunch break is included between completion of i and beginning of j, the corresponding length of time for lunch is not counted in the operator's workday, and the remaining time for the schedule is adjusted accordingly.

Whenever a new schedule is initiated with j, the starting time of the schedule is adjusted to provide time for preparing the vehicle; i.e., schedule begins at $(a_j - pr_k)$, where pr_k is vehicle preparation time at depot k. Recall that knowledge of this starting time is required to specify the breaks appropriately.

If $N' - N_2 = \emptyset$, the procedure stops with a set of reasonable schedules. Otherwise, replace N' by $N' - N_2$. The current set N_1 of source-tasks is used for the next iteration, and a new set N_2 is determined as before.

Another set of reasonable schedules is generated when the subset $\overline{T}_k \subset T_k$ of tasks workable by operators of other depots is non-empty. Indeed, the procedure is reapplied to generate reasonable schedules including only tasks in $T_k - \overline{T}_k$ (i.e., tasks workable only by operators of depot k).

Each reasonable schedule j generated with the procedure is included in Ω^F or Ω^P. Then it is easy to generate the column of problem (P) associated with the corresponding variable y_j. Indeed, $\delta_{ij} = 1$ if and only if task i is included in schedule j. The length of time f_{kj} available for the bulk-task at depot k is obtained by adding up all idle periods between consecutive fixed tasks or between a fixed task and a break which are sufficiently long. Indeed, if an idle period is too short for an operator to initiate any activity, then it should not be included in f_{kj}. Also, $f_{k'j} = 0$ for all $1 \leq k \leq K$, $k' \neq k$. The cost c_j is evaluated as follows:

- the guaranteed minimum number of hours is paid at regular rate

- the additional time exceeding this minimum is paid at overtime rate.

To provide an appropriate number of operators to complete the bulk-task at each depot, special reasonable daily schedules (bulk-schedules) dedicated to this task are generated. Hence, for each depot, for each type of operators allowing for minimal and maximal time length schedule, respectively, we generate several bulk-schedules to execute completely the bulk-task at the depot.

Each bulk-schedule j is included in Ω^F or Ω^P. The elements of the column of problem (P) associated with the corresponding variable y_j are easily generated. First, $\delta_{ij} = 0$ for all $1 \leq i \leq n$, and c_j is evaluated as before. Furthermore, if schedule j is associated with depot k, then $f_{\bar{k}j} = 0$ for all $1 \leq \bar{k} \leq K$, $\bar{k} \neq k$, and f_{kj} is the length of the schedule reduced of the time required for coffee-breaks, preparation and report.

Using all these reasonable generated schedules, we can specify problem (P). The next step of our general approach is to deal with problem (P).

4. SOLVING THE SET-COVERING PROBLEM

In the second step of our general approach, we are dealing with problem (P) using a branch-and-bound method. At each node of the branch-and-bound tree, the lower bound is obtained by solving the corresponding relaxed linear programming problem where integer constraints on variables are deleted. The efficiency of any branch-and-bound method relies on selecting an appropriate branching strategy and on obtaining rapidly a good upper bound on the optimal value to reduce the number of tree nodes enumerated.

4.1 Branching strategies

In general, the classical 0-1 branching strategy generates a left-hand and a right-hand successor nodes by fixing the value of a selected variable to 0 and 1, respectively. Preliminary testing showed the weakness of this strategy in our context. Furthermore, these tests also indicated that, in the optimal solution of the relaxed linear programming problems, many variables were non-integer mainly because of the additional constraints. Indeed, other tests showed that most variables were integer in the optimal solution of the relaxed linear programming problems where these additional constraints were deleted.

For these reasons, we introduce more global branching strategies in the spirit of Nemhauser and Wolsey remarks [8, p. 365]. To specify these four strategies, denote by $\bar{y}$ an optimal solution of the relaxed linear programming problem associated with some node of the branch-and-bound tree:

i) **Strategy with regard to the number of part-time schedules selected.**

If $\sum_{j \in \Omega^P} \overline{y}_j$ not integer, then

- Constraint $\sum_{j \in \Omega^P} y_j \leq \lfloor \sum_{j \in \Omega^P} \overline{y}_j \rfloor$ is added to the left-hand successor,

- Constraint $\sum_{j \in \Omega^P} y_j \geq \lfloor \sum_{j \in \Omega^P} \overline{y}_j \rfloor + 1$ is added to the right-hand successor,

where $\lfloor a \rfloor$ denotes the largest integer smaller or equal to a.

ii) **Strategy with regard to the number of full-time schedules selected.**

If $\sum_{j \in \Omega^F} \overline{y}_j$ not integer, then

- Constraint $\sum_{j \in \Omega^F} y_j \leq \lfloor \sum_{j \in \Omega^F} \overline{y}_j \rfloor$ is added to the left-hand successor,

- Constraint $\sum_{j \in \Omega^F} y_j \geq \lfloor \sum_{j \in \Omega^F} \overline{y}_j \rfloor + 1$ is added to the right-hand successor.

iii) **Strategy with regard to the number of part-time schedules selected for depot k.**

Denote by $\Omega_k^P \subset \Omega^P$ the subset of part-time schedules of operators based at depot k. If $\sum_{j \in \Omega_k^P} \overline{y}_j$ is not integer, then

- Constraint $\sum_{j \in \Omega_k^P} y_j \leq \lfloor \sum_{j \in \Omega_k^P} \overline{y}_j \rfloor$ is added to the left-hand successor,

- Constraint $\sum_{j \in \Omega_k^P} y_j \geq \lfloor \sum_{j \in \Omega_k^P} \overline{y}_j \rfloor + 1$ is added to the right-hand successor.

iv) **Strategy with regard to the number of full-time schedules selected for depot k.**

Denote by $\Omega_k^P \subset \Omega^F$ the subset of full-time schedules of operators based at depot k. If $\sum_{j \in \Omega_k^F} \bar{y}_j$ is not integer, then

- Constraint $\sum_{j \in \Omega_k^F} y_j \leq \lfloor \sum_{j \in \Omega_k^F} \bar{y}_j \rfloor$ is added to the left-hand successor,

- Constraint $\sum_{j \in \Omega_k^F} y_j \geq \lfloor \sum_{j \in \Omega_k^F} \bar{y}_j \rfloor + 1$ is added to the right-hand successor.

These strategies are applied in the order specified above. Whenever iii) or iv) applies, the depot k is selected according to previous experimentation [2] indicating that branching first on sums of variables having a fractional part in the interval [0.4, 0.6] is more efficient.

Finally, whenever none of the above strategies applies, then the 0-1 strategy is used. Now, since our numerical experimentation indicates that the additional constraints (2.2) related to the volume of work due to bulk-tasks at depots are mostly responsible for the fractional variables, then branching is specified using the fractional variable $\bar{y}_j$ with the largest coefficient f_{kj}.

4.2 Node selection

A depth-first search strategy is adopted to identify a feasible solution as rapidly as possible. Whenever one of the branching strategies i) to iv) applies, the left-hand successor is selected if the fractional part of the sum used to specify the strategy is less than or equal to B, $0 \leq B \leq 1$ (B = 0.4 in our implementation). Otherwise, the right-hand successor is selected. Whenever the 0-1 strategy applies, the right-hand successor is selected because, in general, it is more restrictive to fix a variable to 1.

The pending nodes of the branch-and-bound tree are placed in a list in increasing order of the lower bound of their parent. Whenever an integer solution is found, the upper bound is updated and the pending node at the top of the list is selected for the next iteration. On the other hand, whenever the problem is not feasible, then the brother-node is selected next if it is still pending. Otherwise, the pending node at the top of the list is selected.

4.3 Stopping criterion

Given a tolerance factor γ, $0 \leq \gamma \leq 1$, the branch-and-bound procedure stops whenever

$$\frac{UB - LB}{LB} < \gamma.$$

where UB (upper bound) is the value of the best integer solution and LB (lower bound) is the lower bound of the pending node at the top of the list.

4.4 Reducing the feasible domain

As mentioned before, it is important to obtain a good upper bound (UB) as early a possible to reduce the number of nodes enumerated. In this spirit, we introduce additional auxiliary constraints in (P) to reduce the feasible domain, and hence, increasing our chance to identify rapidly an integer solution of (P). To specify these auxiliary constraints, denote by $\bar{y}$ an optimal solution of the relaxed linear programming problem $(\overline{P})$ associated with (P). Consider also some parameter τ, $0 \leq \tau \leq 1$. For all depot k, $1 \leq k \leq K$, so that $\sum_{j \in \Omega_k^f} \bar{y}_j$ is not integer, then add to (P)

$$\text{constraint } \sum_{j \in \Omega_t^r} y_j \geq \lfloor \sum_{j \in \Omega_t^r} \bar{y}_j \rfloor + 1 \quad \text{if } \sum_{j \in \Omega_t^r} \bar{y}_j - \lfloor \sum_{j \in \Omega_t^r} \bar{y}_j \rfloor \geq \tau$$

or

$$\text{constraint } \sum_{j \in \Omega_t^r} y_j \geq \lfloor \sum_{j \in \Omega_t^r} \bar{y}_j \rfloor \quad \text{otherwise .}$$

Denote (P_τ) the problem generated. Of course, the size of the feasible domain of (P_τ) is non-decreasing with the value of τ.

The solution procedure to deal with (P) is to apply the branch-and-bound method described before to a sequence of problems (P_τ) where the value of τ increases (for instance, $\tau = 0$, $\tau = 0.25$, $\tau = 0.5$, $\tau = 0.75$, $\tau = 1$) Note that by convention $(P_1) = (P)$. But since any optimal solution of $(P_{\bar{\tau}})$ is also feasible for any (P_τ) with $\tau \geq \bar{\tau}$, we apply the branch-and-bound method to (P_τ) only if a significant improvement can be expected. Hence, we introduce another parameter $0 \leq \xi \leq 1$, and whenever

$$\frac{UB - v(\bar{P}_\tau)}{v(\bar{P}_\tau)} < \xi$$

($v(\bar{P}_\tau)$ being the optimal value of the relaxed linear programming $(\bar{P}_\tau)$) the expected improvement is said to be too small to apply the branch-and-bound method to (P_τ).

The solution procedure in our implementation can be summarized as follows:

Step 0 Solve $(\bar{P})$. Let $\bar{y}$ be an optimal solution of $(\bar{P})$. If $\bar{y}$ is integer, STOP. Otherwise, let $\tau = 0$ and $UB = \infty$.

Step 1 Solve $(\bar{P}_\tau)$.

$$\text{If } \frac{UB - v(\bar{P}_\tau)}{v(\bar{P}_\tau)} < \xi, \quad \text{go to Step 3.}$$

Step 2 Apply the branch-and-bound method to (P_τ).

Step 3 If $\tau = 1$, STOP.

Otherwise, let $\tau = \tau + 0.25$, and go to Step 1.

5. MODIFYING AND IMPROVING THE SOLUTION

To improve the solution generated with the procedure described in the preceding section, first we look for tasks included in more than one schedule. Each such task is eliminated from all but one schedule including it to generate more time available for the bulk-tasks at corresponding depots.

Secondly, we may share differently the volume of work of the bulk-tasks among the operators in order to reduce total cost. Indeed, since reasonable schedules of minimal and maximal length may have been selected for a depot k, all the time available for the bulk-task in schedules of minimal length must be used up before paying at overtime rate for some portion of schedules of maximal length. Such a minor adjustment may reduce total cost.

6. IMPLEMENTATIONS AND RESULTS

The procedures are implemented in FORTRAN on the Cyber 835 of the University of Montreal. The transportation problems used to generate the reasonable schedules are solved with RNET [6]. Also the XMP library [7] is used to deal with the linear programming problems.

The approach was used to deal with real world problems in the context of postal vehicle crew scheduling. In the example used to test the approach, there are 16 depots, 167 fixed tasks representing approximately 200 hours of work. The bulk-tasks at depots represent approximately 230 hours of work. Among the 167 fixed tasks, 107 can be executed by operators of more than one depot.

Several simulations are analyzed according to different values of the parameters like the lower bound α on the proportion of full-time operators, the minimal length of an idle period in a schedule that can be used to execute part of a bulk-task, and allowing or not, overtime. The number of reasonable schedules generated (i.e., number of variables y_j) varies between 400 and 500. In all these tests, the stopping criterion used is

$$\frac{UB - LB}{LB} < 0.01.$$

Most of the CPU time required is spent on solving the set-covering problem (P) using XMP and the branch-and-bound method. For the example mentioned before, on the average it took

- 20 to 30 seconds to generate the reasonable schedules,
- the time limit (33 minutes) to deal with (P),
- 10 to 20 seconds to modify and improve the solution in the last steps of the approach.

Hence, the approach suffers of the same limitations as other approaches requiring to solve set-covering problems.

We observed an average improvement of 3% over the costs of the solutions generated manually.

Furthermore, a major advantage of the approach is to allow the study of several scenarios when the parameters are modified. To allow any manager to fully exploit the approach, we developed a user friendly computer system embedding the approach and running on PC-type micro-computers.

Finally, note that the solution obtained with the approach could probably be improved by adding a column's generation procedure to obtain additional reasonable schedules that seem interesting with regard to the dual information available when solving (P).

REFERENCES

[1] M. Desrochers, "La fabrication d'horaires de travail pour les conducteurs d'autobus par la méthode de génération de colonnes", Publication # 420, Centre de recherche sur les transports, Université de Montréal (Montréal, 1986).

[2] J. Desrosiers, Y. Dumas, F. Soumis, "The Multiple Vehicle Many to Many Routing Problem with Time Windows", Publication # 362, Centre de recherche sur les transports, Université de Montréal (Montréal, 1986).

[3] J. Desrosiers, J.A. Ferland, J.M. Rousseau, G. Lapalme, L. Chapleau, "TRANSCOL: A Multiperiod School Bus Routing and Scheduling Systems", in: A.J. Swersey and E.J. Ignall, eds., *Studies in the Management Sciences* 22 (North-Holland, Amsterdam, 1986) pp. 47-71

[4] J. Desrosiers, F. Soumis, M. Desrochers, "Routing with Time Window by Column Generation", *Networks* 14 (1984) 545-565.

[5] J.A. Ferland, "The Single Depot Vehicle Problem", in: L. Contesse, R. Correa, A. Weintraub, eds., *Recent Advances in System Modelling and Optimization* (Springer, Berlin, 1986) pp. 99-112.

[6] M. Grigoriadis, T. Hsu, "The Rutgers Minimum Cost Network Flow Subroutines", *SIGMAP* 26 (1979) 17-18

[7] R.E. Marsten, "The Design of XMP Linear Programming Library", *ACM Transactions on Mathematical Software* 7 (1981) 481-497

[8] G.L. Nemhauser, L.A. Wolsey, "*Integer and Combinatorial Optimization*" (Wiley, New York, 1988).

[9] S. Taillefer, "Elaboration des horaires pour les chauffeurs de véhicules postaux", Publication # 557, Centre de recherche sur les Transports, Université de Montréal (Montréal, 1987)